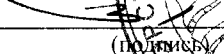


**Министерство образования Республики Беларусь**  
Учебно-методическое объединение по естественнонаучному образованию

**УТВЕРЖДАЮ**

Первый заместитель Министра образования  
Республики Беларусь

  
(подпись) **В.А. Болуш**

« 10 » \_\_\_\_\_ 2014 г.  
Регистрационный № 1Д-Б, 495 /тип.

**МЕТОДЫ ПРОГРАММИРОВАНИЯ И ИНФОРМАТИКА**

**Типовая учебная программа по учебной дисциплине**


**для специальности**

**1-31 03 01 Математика (по направлениям)**

**(1-31 03 01-02 Математика (научно-педагогическая деятельность))**

**СОГЛАСОВАНО**

Начальник Управления высшего  
образования Министерства образования  
Республики Беларусь

  
« 10 » \_\_\_\_\_ 12 \_\_\_\_\_ 2014 г. С.И. Романюк

**СОГЛАСОВАНО**


Председатель  
Учебно-методического объединения по  
естественнонаучному образованию  
Математик

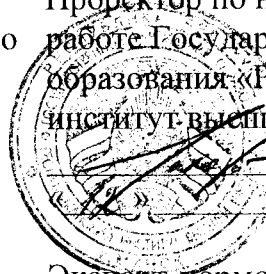
  
(подпись) « 15 » \_\_\_\_\_ 2014 г.



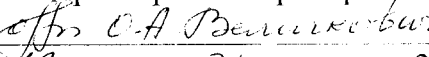
**СОГЛАСОВАНО**

Проректор по научно-методической  
работе Государственного учреждения  
образования «Республиканский  
институт высшей школы»

  
« 12 » \_\_\_\_\_ 11 \_\_\_\_\_ 2014 г. И.В. Титович



Эксперт-нормоконтролер

  
« 12 » \_\_\_\_\_ 12 \_\_\_\_\_ 2014 г.

Минск 2014

Информация об изменениях размещается на сайтах:

<http://www.nihe.bsu.by>

<http://www.edubelarus.info>

**СОСТАВИТЕЛИ:**

Галина Алексеевна Расолько, доцент кафедры веб-технологий и компьютерного моделирования Белорусского государственного университета, кандидат физико-математических наук, доцент;

Елена Васильевна Кремень, доцент кафедры веб-технологий и компьютерного моделирования Белорусского государственного университета, кандидат физико-математических наук, доцент.

Юрий Алексеевич Кремень, доцент кафедры веб-технологий и компьютерного моделирования Белорусского государственного университета, кандидат физико-математических наук, доцент.

**РЕЦЕНЗЕНТЫ:**

Кафедра информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств».

Леонид Александрович Янович, главный научный сотрудник государственного научного учреждения «Институт математики НАН Беларуси», доктор физико-математических наук, член-корреспондент Национальной академии наук Беларуси, профессор;

**РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:**

Кафедрой веб-технологий и компьютерного моделирования механико-математического факультета Белорусского государственного университета (протокол № 6 от 3 февраля 2014 г.);

Научно-методическим советом Белорусского государственного университета (протокол № 4 от 1 апреля 2014 г.).

Научно-методическим советом по математике и механике Учебно-методического объединения по естественному образованию (протокол № 2 от 7 апреля 2014г.)

Ответственный за редакцию: Кремень Елена Васильевна  
Ответственный за выпуск: Кремень Елена Васильевна

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная дисциплина «Методы программирования и информатика» построена на основе обязательного минимума содержания образования по информатике. В неё включаются вопросы, относящиеся собственно к методам программирования: компьютеры, их устройство, системы счисления, представление информации, разработка математических моделей путём соответствующей формализации задач, анализ полученных результатов, запись алгоритмов на алгоритмических языках, и изучаются современные информационные технологии: по обработке текстовой, числовой, графической информации, по хранению, поиску и сортировке информации.

Центральной идеей сквозного образования по методам программирования и информатике на педагогическом потоке является необходимость дать студентам тот набор знаний, который будет полезен им при работе на уроках информатики. Именно этим и определяется выбор программных средств, поэтому в качестве базового языка программирования выбран язык Pascal, имеющий в настоящее время ряд модификаций (Turbo Pascal, Borland Pascal, Free Pascal, Pascal ABC). Только ориентацией на использование в школе обусловлен и выбор математического пакета MathCad, а не более сложных пакетов, например, Mathematica или Maple. MathCad не только использует привычный способ математической записи уравнений, операций и графики, но и проводит все вычисления в автоматическом режиме, что позволяет с максимальной наглядностью и без особых трудностей использовать этот пакет при работе со школьниками.

Второй важнейшей идеей обучения является преимущество полученных знаний.

Третьей важнейшей идеей обучения является необходимость дать инструментarium студенту-математику для проведения трудоёмких расчётов и визуализации математических объектов, возникающих во всех курсах математического цикла.

Задачи учебной дисциплины: овладение приемами и основами методологии структурного, модульного и объектно-ориентированного программирования; изучение фундаментальных алгоритмов и структур данных; выработка творческого подхода к конструированию алгоритмов с целью развития аналитических и творческих способностей студентов; педагогическая и научно-методическая работа в области математики и информационных технологий.

Цели учебной дисциплины: развитие алгоритмического мышления, изучение современных методов программирования, приобретение навыков и освоение работы на современных вычислительных средствах (знакомство с современными методологиями приобретения знаний, приобретение навыков решения разнообразных задач на компьютерах с использованием современных информационных технологий).

На первом этапе предлагается изучать методы программирования на базе языка Turbo Pascal, который позволяет отрабатывать классические приёмы программирования и современные программные технологии, однако, эти же знания можно приобретать и на базе Borland Pascal, Free Pascal, Pascal ABC. Основной упор делается на то, что язык программирования – это средство, позволяющее решить задачу, которую надо предварительно сконструировать, а ещё ранее разработать алгоритм. Разработка алгоритмов является в какой-то мере творческой деятельностью. Однако существует ряд стандартных методов, приёмов и технологий, применяемых для упрощения этого процесса – что составляет предмет техники программирования, владение которой часто облегчает формулирование задач и дальнейшее конструирование эффективных программ их решения.

В этой части изучаются стандартные типы данных языка, управляющие структуры и операторы языка, изучаются вопросы процедурного и модульного программирования, работа с файлами. Основное внимание в курсе уделено не столько вопросу кодирования программы, сколько вопросу проектирования, где упор делается на современные технологии: проектирование сверху-вниз; модульное программирование, т.е. использование аппарата подпрограмм и модулей; проведение анализа эффективности участков программ и их оптимизация; широкое использование аппарата рекурсии. Всё вышеизложенное делается с целью привить некоторый стиль программирования.

Программирование на Turbo Pascal завершается изучением основ методологии объектно-ориентированного программирования (ООП), таких как инкапсуляция, наследование, полиморфизм. Данная тема позволяет обобщить полученные в курсе знания и вывести их на новую более высокую ступень, подготовить студентов к работе с объектно-ориентированными языками программирования Visual Basic, Delphi (Object Pascal), C++, которые они будут изучать в дальнейшем.

На втором этапе предлагается изучать фундаментальные алгоритмы и структуры данных. Здесь отрабатываются навыки алгоритмизации задач по обработке структур данных, в том числе связанных динамических структур: списков, стеков, очередей и деревьев. Из изучаемых фундаментальных алгоритмов отметим алгоритмы сортировки данных, поиска элемента, широкий спектр логико-комбинаторных задач. Алгоритмы сортировок – идеальный объект для иллюстрации многих принципов программирования и ситуаций, встречающихся в других задачах. Важнейшей частью изучения этой темы является сравнение алгоритмов сортировки по нескольким критериям, проведение математического анализа этих алгоритмов с последующей проверкой на компьютере для типовых модельных задач. Анализ алгоритмов сортировки важен для правильного выбора способа решения поставленной задачи, учитывающего особенности конкретной задачи, а также для развития аналитических и творческих способностей студентов.

На третьем этапе предлагается изучать математический пакет MathCad. Будущие педагоги знакомятся с возможностями символьных и численных вычислений, визуализацией результатов на примере решения как задач элементарной математики, что позволит им в дальнейшем применить накопленный опыт в школе, так и задач из линейной алгебры, дифференциальных уравнений, численного анализа. Основной упор делается не на использование стандартных возможностей пакета, хотя и этому уделяется достаточное внимание, а на написание собственных программ различных алгоритмов в MathCad, посредством встроенного языка программирования.

Каждый обучающийся имеет возможности закрепить лекционный материал на лабораторных занятиях и занятиях по вычислительной практике. При этом значительно возрастает роль самостоятельной работы студентов над предметом, без чего успешное освоение предмета представляется маловероятным. Индивидуальные задания содержат расширенный спектр задач по обработке различных данных, написанию процедур, функций, модулей с учетом методологий разработки больших программ, что позволяет отработать стандартные приемы работы с различными устройствами (клавиатурой, курсором, звуком, работа в текстовом и графическом видео режимах) и прочее.

Программа составлена с учетом межпредметных связей и программ по смежным дисциплинам: введение в математику, аналитическая геометрия, алгебра и теория чисел, математический анализ, дифференциальные уравнения.

***В результате изучения учебной дисциплины студент должен знать:***

- методы решения научно-технических и информационных задач;
- современные информационные технологии;

***уметь:***

- решать типовые задачи математики и информатики;
- работать на современных вычислительных средствах;
- применять современные информационные технологии и методы реализации решения прикладных задач.

***владеть:***

- навыками работы с современными информационными технологиями;
- навыками программирования алгоритмов обработки данных;
- методами программирования задач в различных областях;
- современными технологиями разработки программ.

Учебный курс рассчитан на 468 часов, из них 280 аудиторных часов, в том числе 140 часов лекций, 140 часов лабораторных занятий на персональных компьютерах.

## ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

Учебный материал курса «Методы программирования и информатика» разделен на несколько базовых разделов, каждый из которых содержит ряд тем, рассматриваемых в ходе лекций и на лабораторных занятиях.

№ раз- делов и тем	Наименование разделов и тем	Количество часов	
		Лекции	Лабораторные занятия
<b>1.</b>	<b>Теория и практика программирования на языке Pascal</b>	<b>82</b>	<b>82</b>
1.1.	Введение. Арифметика ЭВМ	2	6
1.2.	Алгоритмизация	6	6
1.3.	Средства алгоритмического языка Pascal	2	-
1.4.	Введение в систему типов	2	-
1.5.	Простые данные языка Pascal и работа с ними	6	4
1.6.	Элементарные средства по работе с данными	4	4
1.7.	Базовые операторы языка и методы программирования	4	6
1.8.	Структуры данных и работа с ними средствами алгоритмического языка	4	4
1.9.	Механизмы структурирования программ	8	8
1.10.	Файлы в языке Pascal	10	10
1.11.	Специальные средства языка Pascal	8	8
1.12.	Стандартные приемы работы с устройствами IBM-PC	6	6
1.13.	Графическое программирование	8	8
1.14.	Методологии разработки больших программных комплексов	2	2
1.15.	Особенности работы с различными диалектами и реализациями языка Pascal	10	10
<b>2.</b>	<b>Отображение абстрактных структур данных на структуры данных хранения. Основные операции над ними</b>	<b>44</b>	<b>44</b>
2.1.	Технологии реализации алгоритмов	10	10
2.2.	Сортировка данных	10	10
2.3.	Динамические структуры данных	8	8
2.4.	Задачи поиска элемента в разных структурах данных	2	2
2.5.	Методы разработки алгоритмов	14	14
<b>3</b>	<b>MathCad</b>	<b>14</b>	<b>14</b>
3.1.	Характерные черты MathCad	2	-
3.2.	Архитектура MathCad и его интерфейс	2	-
3.3.	Средства редактирования	2	2
3.4.	Векторы и матрицы	2	2
3.5.	Встроенные операторы и функции	2	2
3.6.	Программирование	2	2
3.7.	Решение уравнений и систем	2	2
3.8.	Файлы данных	-	2
3.9.	Анимация	-	2
<b>ИТОГО</b>		<b>140</b>	<b>140</b>

## СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

### **Раздел 1. Теория и практика программирования на языке Pascal.**

#### *Тема 1.1. Введение.*

Понятие о программировании, как о науке. Ключевые положения. Развитие языков программирования. Эволюция языка Pascal. Система программирования Turbo Pascal (Free Pascal, Pascal ABC). Интегрированная среда. Упрощенная модель компилятора.

Арифметика ЭВМ. Системы счисления. Формы представления данных.

#### *Тема 1.2. Алгоритмизация.*

Алгоритмы. Основные этапы решения задач на ЭВМ. Способы представления алгоритмов. Блок-схемы. Структурограммы. Алгоритмические языки. Структурное программирование и точность программ. Структурные блок-схемы. Основные конструкции структур управления. Структуры управления и их реализация в программе.

Методы разработки алгоритмов и программ: проектирование сверху-вниз, снизу-вверх, модульное программирование, структурное кодирование. Тестирование и отладка алгоритмов и программ.

#### *Тема 1.3. Средства алгоритмического языка Pascal.*

Общая характеристика алгоритмических языков. Понятие синтаксиса и семантики языка программирования. Способы формального описания языков программирования на базе языка Pascal.

Базовые элементы языка Pascal. Набор символов, лексемы, разделители. Общая структура Pascal программы.

#### *Тема 1.4. Введение в систему типов.*

Типы данных. Константы и переменные. Абсолютные константы.

Система типов: их описание, реализация в памяти ПК, действия с данными заявленного типа.

#### *Тема 1.5. Простые данные языка Pascal и работа с ними.*

Схема простых типов. Целочисленные типы. Битовая арифметика. Вещественные типы. Выражения. Приоритет операций. Символьный, булевский, адресный, перечислимый тип данных. Диапазоны.

#### *Тема 1.6. Элементарные средства по работе с данными.*

Присваивание значений данным. Простейшее описание процедур и функций. Знакомство с файловой системой. Текстовые стандартные файлы. Ввод данных разных типов. Вывод данных разных типов.

#### *Тема 1.7. Базовые операторы языка и методы программирования.*

Классификация операторов.

Простые операторы. Оператор присваивания. Совместимость по присваиванию. Оператор безусловного перехода, пустой оператор, составной оператор.

Структурные операторы. Условный оператор. Методы и приемы программирования. Оператор варианта. Операторы повторения. Программиро-

вание циклов с известным числом повторений, циклов с предусловием, с постусловием. Обработка последовательностей. Итерационные алгоритмы высшей математики.

*Тема 1.8. Структуры данных и работа с ними средствами алгоритмического языка.*

Проблемы отображения Абстрактных Структур Данных на Структуры Данных Хранения.

Порядковые типы. Массивы данных, типизированные константы-массивы. Действия над элементами массива.

Строковый тип. Редактирование строк при помощи встроенных подпрограмм.

Множественный тип, представление, действия, реализация.

Записи – простейшее описание. Оператор присоединения.

Изменение/приведения типов и значений.

*Тема 1.9. Механизмы структурирования программ.*

Полное описание процедур и функций. Параметры. Принцип локализации. Побочный эффект. Рекурсии и итерации. Процедурные типы. Переменные процедурных типов. Программирование алгоритмов с использованием подпрограмм. Модули. Модули пользователя. Эффективность программ.

*Тема 1.10. Файлы в языке Pascal.*

Схема типов языка. Записи. Записи с вариантами.

Файлы. Физические модели. Алгоритмическая реализация. Файловые типы. Типизированные файлы. Операции над файлами. Алгоритмы работы с файлами: создание, корректировка, чтение, обработка ошибок ввода-вывода. Текстовые файлы. Специфика обработки. Нетипизированные (бинарные) файлы. Ввод-вывод простых и структурированных данных.

*Тема 1.11. Специальные средства языка Pascal.*

Модуль System. Динамическая память. Указатели, ссылочные данные. Средства для работы с адресами. Размещение и освобождение динамических переменных. Управление состоянием Heap, анализ состояния Heap. Программирование алгоритмов с использованием указателей. Работа с динамическими массивами: одномерными и двумерными. Введение в связанные динамические структуры данных: линейные списки, стеки, очереди, деревья.

*Тема 1.12. Стандартные приемы работы с устройствами IBM-PC.*

Основные положения. Работа с клавиатурой, звуком. Управление курсором. Работа в текстовом видеорежиме. Текстовые окна. Видеодоступ.

*Тема 1.13. Графическое программирование.*

Основные положения. Базовые процедуры и функции.

Управление параметрами изображений.

Построение графических примитивов.

Работа с текстом.

Экран и окно.

Манипулирование фрагментами образов. Анимация.



*Тема 1.14. Методологии разработки больших программных комплексов.*

Структурная методология разработки программ. Цели, основные принципы структурной методологии, модульное программирование.

Объектно-ориентированная методология разработки программ (ООП). Простейшие положения. Иерархия объектов. Механизмы объявления объектов. Механизмы определения методов.

*Тема 1.15. Особенности работы с различными диалектами и реализациями языка Pascal.*

Версии реализации и среды разработки Borland Pascal, Free Pascal, Pascal ABC.

**Раздел 2. Отображение абстрактных структур данных на структуры данных хранения. Основные операции над ними.**

*Тема 2.1. Технологии реализации алгоритмов.*

Технологии структурного программирования. Основные принципы. Важность защиты от логических ошибок. Этапы создания структурной программы: постановка задачи, выбор модели и метода решения задачи, разработка внутренних структур данных, проектирование, структурное программирование, тестирование, правила программирования.

Объектно-ориентированные средства языка Pascal. Основные принципы: инкапсуляция, наследование, полиморфизм.

Описание объектов. Поля и методы. Виртуальные и динамические методы. Наследование. Экземпляры объектов.

Совместимость объектных типов. Совместимость между экземплярами объектов. Совместимость между указателями на экземпляры объектов. Совместимость между формальными и фактическими параметрами.

Динамические и виртуальные объекты. Конструкторы. Освобождение объектов. Деструкторы. Обработка ошибок при работе с динамическими объектами.

Объекты и модули.

*Тема 2.2. Сортировка данных.*

Внутренняя и внешняя сортировки. Сортировка массивов «in situ». Классы алгоритмов сортировок. Базовые алгоритмы. Улучшенные алгоритмы. Характеристики методов сортировки, анализ их ресурсов.

Сортировка обменом (метод простого обмена и последующие улучшения, шейкер сортировка, быстрая сортировка). Сортировка включениями (сортировка простым включением, сортировка простым включением с бинарным поиском места включения, сортировка Шелла). Сортировка выбором (сортировка простым выбором, пирамидальная сортировка).

Сортировка массивов за линейное время. Карманная сортировка. Сортировка методом подсчета.

Сортировка последовательных файлов. Простое слияние. Естественное слияние. Сбалансированное многопутевое слияние. Улучшенные методы: многофазная и каскадная сортировка.

*Тема 2.3. Динамические структуры данных.*

Абстрактные типы данных. Общие сведения о динамических структурах данных.

Списки и их классификация. Работа с данными типа список. Однонаправленные и двунаправленные списки. Сортировка элементов списка. Слияние упорядоченных списков. Практическая работа с данными типа список.

Стеки. Решение задач с использованием стеков.

Очереди. Очереди с приоритетом. Очереди с двухсторонним доступом. Решение задач с использованием очереди.

Деревья. Основные понятия и определения. Представление дерева последовательным и связным методом хранения. Бинарные деревья. Создание идеально сбалансированного дерева. Создание дерева поиска. Обход дерева. Решение задач с использованием бинарных деревьев.

Сильно ветвящиеся деревья. Решение задач с использованием сильно ветвящихся деревьев.

*Тема 2.4. Задачи поиска элемента в разных структурах данных.*

Последовательный поиск в массиве, в списке, в дереве.

Бинарный поиск в массиве, в списке, в дереве поиска.

Вставка элемента в массив, в список, в дерево.

Уничтожение элемента в массиве, в списке, в дереве.

Алгоритмы поиска медианы последовательности.

*Тема 2.5. Методы разработки алгоритмов.*

Алгоритмы типа «разделяй и властвуй». Рекурсивные и не рекурсивные алгоритмы. Задача о ханойских башнях: рекурсивная реализация и реализация при помощи дерева. Задача на получение перестановки чисел (рекурсивный и не рекурсивный варианты решения).

Алгоритмы с возвратом. Задача обхода конем шахматной доски. Задача о восьми ферзях. Задача о восьми ладьях.

«Жадные» алгоритмы. Задача на счастливые билеты. Решение задачи получения сдачи эвристическим методом и методом полного перебора всех возможных вариантов рекурсивным методом и с помощью сильно ветвящегося дерева.

Поиск с возвратом и локальный поиск. Задача о рюкзаке на множестве положительных чисел. Задача о рюкзаке на множестве вещественных чисел.

Фракталы. Алгебраические, геометрические, стохастические фракталы. Фрактальные множества: множества Мандельброта и множества Жулиа. Модели магнетизма. Ньютоновские фракталы. Примеры стохастических фракталов – папоротник. Конструктивные фракталы: множества Кантора, фракталы Серпинского, Гильберта и другие.

### **Раздел 3. MathCad.**

#### *Тема 3.1. Характерные черты MathCad.*

Интерфейс, численные методы, символьные подсчеты, построение графиков, характеристики текста и рабочего документа, особенности системы программирования.

#### *Тема 3.2. Архитектура MathCad и его интерфейс.*

Меню, палитры, простые просчеты, ввод текста, области, определение переменных, дискретные переменные, глобальные переменные, определение функций, таблицы, форматирование таблиц, построение графиков и поверхностей, справка, электронные книги, редактирование формул.

#### *Тема 3.3. Средства редактирования.*

Выделение и копирование области, защита подсчетов, руководство подсчетами, форматирование выражений и значений, математические шрифты.

#### *Тема 3.4. Векторы и матрицы.*

Создание, заполнение, нижние и верхние индексы. Присваивание массивам значений, удаление/вставка элемента.

#### *Тема 3.5. Встроенные операторы и функции.*

Подсчет сумм, произведений, производных, интегралов. Пользовательские операторы. Кусочно-непрерывные функции (if, until).

#### *Тема 3.6. Программирование.*

Условная передача управления, операторы циклов, использование подпрограмм и рекурсий.

#### *Тема 3.7. Решение уравнений и систем.*

Численное и символьное решение алгебраических уравнений и систем уравнений или неравенств. Решение дифференциальных уравнений. Особенности символьной обработки.

#### *Тема 3.8. Файлы данных.*

Функции записи/считывания скалярных данных и матриц.

#### *Тема 3.9. Анимация.*

Создание анимации, воспроизведение анимационного клипа.

## ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

### ЛИТЕРАТУРА

#### Основная

1. Расолька, Г.А. Pascal: тэорыя і практыка праграміравання: вучэб.-метадыч. дапам. / Г. А. Расолька, Ю. А. Кремень. – Мн.: БДУ, 2008.
2. Расолька, Г.А. Метады праграміравання. Алгарытмы апрацоўкі даных / Г. А. Расолька, Ю. А. Кремень. . – Мн.: БДУ, 2008.
3. Расолька, Г.А. Использование информационных технологий в курсе вузовской математики. В 3-х частях. Часть 1. Решение задач в пакете MathCad. Учеб.-метод. пособие / Г. А. Расолька, Ю. А. Кремень, Н. В. Бровка, Л. Г. Третьякова. - Мн.: БГУ, 2010.
4. Расолька, Г.А. Использование информационных технологий в курсе вузовской математики. В 3-х частях. Часть 2. Решение задач в пакетах MathCad и Mathematica. Учеб.-метод. пособие / Г. А. Расолька, Е. В. Кремень, Ю. А. Кремень, Л. Г. Третьякова. - Мн.: БГУ, 2011.
5. Расолька, Г. А. Использование информационных технологий в курсе вузовской математики. В 3-х частях. Часть 3. Решение задач аналитической геометрии в пакете MathCad. Учеб.-метод. пособие / Г. А. Расолька, Ю. А. Кремень. - Мн.: БГУ, 2012.
6. Расолька, Г. А. Метады праграміравання і інфарматыка: канспект лекцый для студэнтаў мех.-мат. фак. спец. 1-31 03 01-02 «Матэматыка (навукова-педагагічная дзейнасць)». У 2 ч. Ч. I: Мова Pascal / Г. А. Расолька, Е. В. Крэмень, Ю. А. Крэмень. – Мінск: БДУ, 2013. <http://elib.bsu.by/handle/123456789/32342>
7. Расолька, Г. А. Метады праграміравання і інфарматыка. Канспект лекцый для студэнтаў мех.-мат. фак. спец. 1-31 03 01-02 «Матэматыка (навукова-педагагічная дзейнасць)». У 2 ч. Ч. II: Алгарытмы апрацоўкі даных. / Г. А. Расолька, Е. В. Крэмень, Ю. А. Крэмень. – Мінск: БДУ, 2013. <http://elib.bsu.by/handle/123456789/32343>
8. Расолька, Г. А. Зборнік задач па курсу метады праграміравання і інфарматыка: дапаможнік для студэнтаў мех.-мат. фак. спец. 1-31 03 01-02 «Матэматыка (навукова-педагагічная дзейнасць)». У 2 ч. Ч. I. Асноўныя прыёмы праграміравання / Г. А. Расолька, Е. В. Крэмень, Ю. А. Крэмень. – Мінск: БДУ, 2013. <http://elib.bsu.by/handle/123456789/32344>
9. Расолька, Г. А. Зборнік задач па курсу метады праграміравання і інфарматыка: дапаможнік для студэнтаў мех.-мат. фак. спец. 1-31 03 01-02 «Матэматыка (навукова-педагагічная дзейнасць)». У 2 ч. Ч. II. Алгарытмы апрацоўкі даных. Вучэбна-метадычны дапаможнік / Г. А. Расолька, Е. В. Крэмень, Ю. А. Крэмень. – Мінск: БДУ, 2013. <http://elib.bsu.by/handle/123456789/32345>
10. Расолька, Г. А. Заданні вылічальнай практыкі па курсу «Метады праграміравання і інфарматыка»: дапаможнік для студэнтаў мех.-мат. фак.

- спец. 1-31 03 01-02 «Матэматыка (навукова-педагагічная дзейнасць)» / Г. А. Расолька, Е. В. Крэмень, Ю. А. Крэмень. – Мінск: БДУ, 2013. <http://elib.bsu.by/handle/123456789/32346>
11. Расолько, Г. А. Учебно-методический комплекс по учебной дисциплине «Методы программирования и информатика» для специальности 1-31 03 01-02 Математика (научно-педагогическая деятельность) / Г. А. Расолько, Е. В. Кремень, Ю. А. Кремень. - Минск: БГУ, 2013. <http://elib.bsu.by/handle/123456789/32349>
  12. Расолько, Г. А. Методы программирования и информатика. MathCad. Краткий справочник с заданиями : пособие для студентов мех.-мат. фак. спец. 1-31 03 01-02 «Математика (научно-педагогическая деятельность)» / Г. А. Расолько, Е. В. Кремень, Ю. А. Кремень. – Мінск: БГУ, 2013. <http://elib.bsu.by/handle/123456789/36753>

#### Дополнительная:

1. Аляев, Ю. А. Практикум по алгоритмизации и программированию на языке Pascal: учеб. пособие / Ю. А. Аляев, В. П. Гладков, О. А. Козлов. М. : Финансы и статистика, 2004.
2. Ахо, А. В. Структуры данных и алгоритмы : учеб. пособие / А. В. Ахо, Д. Хопкрофт, Д. Д. Ульман. М. : Вильямс, 2000.
3. Бакнелл, Дж. Фундаментальные алгоритмы и структуры данных в Delphi / Дж. Бакнелл. СПб. : ДиаСофтЮП, 2003.
4. Васин, В. В. Элементы нелинейной динамики от порядка к хаосу / В. В. Васин. Екатеринбург : Изд-во Уральского ун-та, 2003.
5. Вирт, Н. Алгоритмы + структуры данных = программы / Н. Вирт. М. : Мир, 1985.
6. Джулиан, М. Фундаментальные алгоритмы и структуры данных в Delphi / М. Джулиан. СПб. : ДиаСофтЮП, 2003.
7. Кетков, Ю.Л. Свободное программное обеспечение Free Pascal для студентов и школьников / Ю.Л. Кетков, А.Ю. Кетков. – СПб.: БХВ-Петербург, 2011.
8. Кнут, Д. Искусство программирования : в 3 т. / Д. Кнут. М. : Вильямс, 2000. Т. 3 : Сортировка и поиск.
9. Кормен, Т. Алгоритмы. Построение и анализ / Т. Кормен [и др.]. 2-е изд. М. : Вильямс, 2005.
10. Кроновер, Р. М. Фракталы и хаос в динамических системах / Р. М. Кроновер. М. : Постмаркет, 2000.
11. Мандельброт, Б. Фрактальная геометрия природы / Б. Мандельброт. М. : Мир, 2002.
12. Маркушевич, А. И. Замечательные кривые / А. И. Маркушевич. М. : Наука, 1978.
13. Морозов, А. В. Введение в теорию фракталов / А. В. Морозов. М., Ижевск

: Ин-т компьютерных исслед., 2002.

14. Немнюгин, С. А. Turbo Pascal : практикум / С. А. Немнюгин. СПб. : Питер, 2003.
15. Никулин, Е. А. Компьютерная геометрия и алгоритмы машинной графики / Е. А. Никулин. СПб. : БХВ-Петербург, 2005.
16. Павловская, Т. А. Pascal. Программирование на языке высокого уровня / Т. А. Павловская. СПб. : Питер, 2004.
17. Пайтген, Х.-О. Красота фракталов. Образы комплексных динамических систем / Х.-О. Пайтген, П. Х. Рихтер. М. : Мир, 1993.
18. Программирование на языке Паскаль: задачник / под ред. О. Ф. Усковой. СПб. : Питер, 2002.
19. Ускова, О. Ф. Программирование алгоритмов обработки данных / О. Ф. Ускова. СПб. : БХВ-Петербург, 2003.
20. Фаронов, В. В. Турбо Паскаль 7.0. Начальный курс / В. В. Фаронов. М. : Нолидж, 1997.

## **МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ**

Организация обучения студентов, основанная на возрастании роли самостоятельной работы студентов, и комплексное методическое обеспечение управляемой самостоятельной работы являются важнейшими задачами высшего образования. Существенную роль по техническому сопровождению учебного процесса играют новые информационные технологии типа сетевых образовательных платформ (СОП). Рассмотрим организацию работы, например, на базе СОП e-University. Это комплекс программных средств, обеспечивающих возможность дистанционного обучения студентов. Для студентов дневной формы обучения СОП e-University обеспечивает получение доступа к заданиям по лабораторным и практическим занятиям, лекционным материалам, предоставляет возможность прохождения тренировочных и контрольных тестирований, как по отдельным темам, так и итоговых и экзаменационных тестирований.

Компьютерное тестирование обеспечивает индивидуальный контроль знаний, регулярность его проведения, полную, объективную проверку знания учебного материала, единство требований, соответствие международным, государственным, вузовским стандартам. Экзамен в форме компьютерного тестирования позволяет проверить знания по большинству вопросов дисциплины, обеспечивает достаточно высокую надежность оценки знаний, освобождает преподавателей от трудоемкой проверки письменных работ.

Организация управляемой самостоятельной работы студентов складывается из нескольких этапов. На первом этапе были выделены те разделы учебной программы, которые вынесены на самостоятельную проработку. На втором этапе проведено структурирование материала: определены учебные модули, последовательность их изучения и контролирующие мероприятия, завершающие изучение каждого модуля. На третьем этапе определен вес каждого модуля в формировании успешности освоения курса в целом и разработаны критерии оценок работы студентов. На четвертом этапе разработано соответствующее методическое обеспечение.

На закрепление теоретического материала вынесено решение задач в среде Pascal (I курс), Object Pascal, Excel, Mathcad (II курс). Для контроля самостоятельной работы студентов примерно раз в три-четыре недели проводится тестирование в СОП e-University. Студент имеет возможность пройти пробный, тренировочный тест перед сдачей каждого теста и проанализировать свои ответы. Если это контролирующий тест, то и преподаватель имеет возможность просмотреть результаты тестирования, как по отдельным студентам, так и по всей группе. Система оценивает решение каждого задания по заранее заложенным весовым коэффициентам и по окончании тестирования всей группы каждому студенту выставляется рейтинговая оценка за

пройденный тест. В конце семестра подводится итог самостоятельной работы – итоговый тест. Его результаты – это либо предварительная оценка на экзамене, либо зачётная (незачётная) оценка. На экзамене проводится экзаменационное тестирование, как составляющая часть общей оценки (около 60%).

Следует подчеркнуть, что объективность, надежность оценки знаний, другие преимущества компьютерного тестирования могут быть реализованы лишь при соответствующем содержании тестовых заданий и качестве тестов.



## **РЕКОМЕНДОВАННЫЕ СРЕДСТВА ДИАГНОСТИКИ РЕЗУЛЬТАТОВ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ**

Рекомендуются следующие формы диагностики компетенций.

### **Устная форма**

1. Собеседование
2. Коллоквиумы
3. Доклады на семинарских занятиях

### **Письменная форма**

1. Контрольные опросы
2. Контрольные работы

### **Устно-письменная форма**

1. Отчеты по лабораторным работам с их устной защитой
2. Зачеты
3. Экзамены

### **Техническая форма**

1. Электронные тесты